

Input Space Partitioning Demo

Dr. Kosta Damevski

CMSC 425/525 - Intro to Software Analysis and Testing
Spring 2023



VCU

Computer Science
College of Engineering

Review of In-Class Exercises

Last Time

- Input Space Partitioning focuses on the input to the method
 - Partitions the input space based on different characteristics
 - Criteria combines characteristics (all pairs, base choice, etc.)
 - Resulting test requirements are mapped to individual test cases
- Today: use an example to further illustrate this

Test Target

- Java's `Iterator` interface
 - <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Iterator.html>
- Three methods:
 - `boolean hasNext()`
 - `E next()`
 - `void remove()` throws `UnsupportedOperationException`, `IllegalStateException`
- Note that there is also an implicit parameter - the current element the iterator is pointing to

Iterator In Action

```
private static void printIterator(Iterator<String> iter)
{
    while(iter.hasNext()) {
        String element = (String) iter.next();
        System.out.print(element + "\n");
    }
}

public static void main(String[] args)
{
    ArrayList<String> list = new ArrayList<>();

    list.add("apple");
    list.add("pie");

    Iterator iter = list.iterator();
    printIterator(iter);

    Iterator iter2 = list.iterator();
    iter2.next();
    iter2.remove();
    printIterator(iter2);
}
```

Output:

=====

apple

pie

pie

Demo

- Iterator in action

Input Domain Modeling

- Two approaches:
 - Interface-based and functionality-*based*
- Our task: come up with a few characteristics for input-domain testing the `Iterator` interface
 - Hint: use the `Iterator` state

One Set of Characteristics

- C1: iterator has more values (true/false)
- C2: iterator returns a non-null object (true/false)
- C3: does not throw UnsupportedOperationException (true/false) — i.e., remove is supported and removes objects
- C4: does not throw IllegalStateException (true/false) — i.e., remove constraint is satisfied

one set of possibilities, many others are possible

One Set of Characteristics

mapped to methods

- hasNext()
- next()
 - C1: iterator has more values (true/false)
 - C2: iterator returns a non-null object (true/false)
- remove()
 - C3: remove is supported and removes objects - does not throw UnsupportedOperationException (true/false)
 - C4: remove constraint is satisfied - does not throw IllegalStateException (true/false)

Criteria to Combine Characteristics

- Our task: Think about which criteria we discussed makes the most sense
 - All Pairs; Pair-wise; T-wise; Base choice; Each choice

Base Choice Coverage for C1-C4

- Base Case (“happy” case) - TTTT
 - iterator has values & returns not null & remove is supported & constraint is satisfied
- Remaining combinations — one criteria is false, while remainder stay true
 - TTTT; FTTT; TFTT; TTFT; TTTF
 - FTTT is not feasible

Demo Implementation

Your Turn

- Add additional tests to:
 - cover cases that are poorly covered (e.g. FFTT?) and/or
 - refactor testing code

Assignment 1

- Discussion

References

- “Introduction to Software Testing” 2nd Edition.
Ammann and Offutt